APPENDIX C

Chapter 6

NetCentric and Client/Server Computing

Netcentric and Client/Server Computing

A Practical Guide

1999

Mark Goodyear
Hugh W. Ryan
Scott R. Sargent
Stanton J. Taylor
Timothy M. Boudreau
Yannis S. Arvanitis
Richard A. Chang
John K. Kaltenmark
Nancy K. Mullen
Shari L. Dove
Michael C. Davis
John C. Clark
Craig Mindrum

Andersen
Consulting

Library of Congress Cataloging-in-Publication Data

Netcentric and client/server computing : a practical guide
Andersen Consulting.
p. cm.
Includes index.
ISBN 0-8493-9967-X (alk. paper)
1. Netcentric computing. I. Andersen Consulting.
QA76.9.N47N48 1998
004.6--dc21
98-44476
CIP

Direct all inquiries to CRC Press LLC, 2000 Corporate Blvd., N.W., Boca Raton, Florida, 33431.

Auerbach is an imprint of CRC Press LLC
No claim to original U.S. Government works
International Standard Book Number 0-8493-9967-X
Printed in the United State of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Table of Contents

SECTION I
OVERVIEW OF NETCENTRIC COMPUTING SOLUTIONS

SECTION II
ARCHITECTURES AND FRAMEWORKS FOR NETCENTRIC COMPUTING

SECTION III
DESIGNING AND IMPLEMENTING NETCENTRIC SOLUTIONS

A French proverb runs, "Plus ça change, plus c'est la même chose": "the more things change, the more they remain the same." We are fortunate that, in the midst of great technology and business change, there are a set of underlying principles of technology that help us decipher the clues hidden in the latest jargon, and also help us create frameworks to navigate the ever-changing route toward greater business results. As implementers of solutions, we must strive to balance the often-conflicting demands that the ndividual parts of the system impose on the overall design. As those parts grow in number and complexity, as they do with netcentric, our juggling acts must become even more professional as we take the pragmatic steps owards that business vision.

Can a single book guarantee you success? Certainly not. Individual success depends on many factors including insight, timing, hard work, experience and a commitment to quality. It is the same with cooking — having the ecipe of the best chef in the world does not mean you will make a living in he restaurant business. We try here to contribute to your recipe book with ur understanding and insight of implementing new technology. The timng, hard work, and commitment is up to you. My father once told me "A vise person learns from experience, a wiser person learns from other peole's experiences."

ll us wise folk can make you a bit wiser on the steps to your own successes, then our writing and your reading have been worthwhile.

Mark Goodyear, Andersen Consulting

Acknowledgments

This book represents the accumulated knowledge and experience of many individuals and groups within Andersen Consulting, in addition to the named authors. We would like to thank, in particular, the following people who made substantive contributions to the information in this book:

Michael W. Alber
Prem N. Mehra
William M. Gilliland
Anthony Roby
Chris M. Degiorgio
David A. Antoniolli
David K. Black
James P. Behling
William W. Westerman
Anatole V. Gershman
Glover T. Ferguson
David K. Black
Jennifer L. Wilson
John P. Jordan
Sharon K. Dietz
Bill G. Chang
David Chen
Jeff A. Goldberg
Stephen A. Barlock

Chapter 6
Communications Architectures

Netcentric computing is an outgrowth of the increased importance and capabilities of the network as well as the associated communications that the network enables between computing environments and access devices. Netcentric computing implies being connected anywhere, anytime. It transforms the common phrase "The network is the computer" into "The network is everywhere." The popular and academic journals, professional gatherings, and our firm's work with organizations all bear witness to the fact that people have now realized the scope of communications issues in developing business solutions today.

In netcentric computing, the network is no longer simply a pipe that moves data from point A to point B. Instead, networks provide a rich set of services that are increasingly more sophisticated to keep pace with the requirements of netcentric applications.

This chapter explores an architectural framework that categorizes and defines the services that are provided by the network and discusses areas in which these services are evolving. Specific networking technologies that are evolving to support netcentric applications are discussed in Section III of this book.

WHAT IS A COMMUNICATIONS ARCHITECTURE?

The evolving role of a network can be seen in the advent of such concepts as "electronic commerce" and the "virtual enterprise." The network continues to support traditional types of data traffic in an individual corporate enterprise, (i.e., local area networks (LANs) and wide area networks (WANs). However, nontraditional business flows (video, graphics, voice, etc.) also need to be supported as well as the new relationships created in the virtual enterprise. Companies that produce the final packaged product or service interact with their suppliers through a seamless information infrastructure. In addition, the need to support an ever-increasing base of public access from home and mobile locations further stretches and redefines the old network boundaries. As computing becomes more distributed and pervasive, the role of the network will grow to support and enable this exchange of content between any point that generates or uses information. Exhibit 1 illustrates these key characteristics of the network.

What is a network? The domain of the network may be defined as the portion of the overall enterprise technology architecture that supports the movement of knowledge in a digital, electronic format between different locations. To provide this capability, the network is composed of communications hardware, software, and services. The network does not include the computing platforms, knowledge technologies, or business logic and applications. However, all network components must provide well-defined services and interfaces to interact effectively with these other technology components.

COMMUNICATIONS ARCHITECTURE GUIDING PRINCIPLES

How is this definition of a network any different from the way the role of the network has been perceived in the past? At first glance, the answer may be, "Not much." However, some fundamental concepts are introduced here that are key to how network architectures will be viewed in the foreseeable future. While the following guiding principles affect all aspects of near-term computing architectures, they will have some very specific impacts to the network domain that will guide the characteristics of future communication architectures.

Netcentric Computing

A netcentric architecture is a standard architecture that allows internal users, customers, and business partners to use multiple electronic access devices (e.g., PCs, mobile computers, kiosks, telephones, etc.) to access disparate sources of information. Netcentric architectures employ open, commonly accepted standards for the network, client, and associated components (e.g., Internet, TCP/IP, Web Browser, ActiveX, COM, CORBA, Java, etc.). Web-based solutions are examples of netcentric architectures. A netcentric architecture requires an intelligent, flexible, standards-based network.

Individuals

The physical network infrastructure is shifting to support more dynamic human-to-human communications styles instead of the traditional, precise computer-to-computer communications. Until now, application requirements were the sole driver for network designs. Now, with interaction styles mimicking more human traits, networks must incorporate multimedia, workflow, collaboration, and other qualities that better support how different individuals use the network.

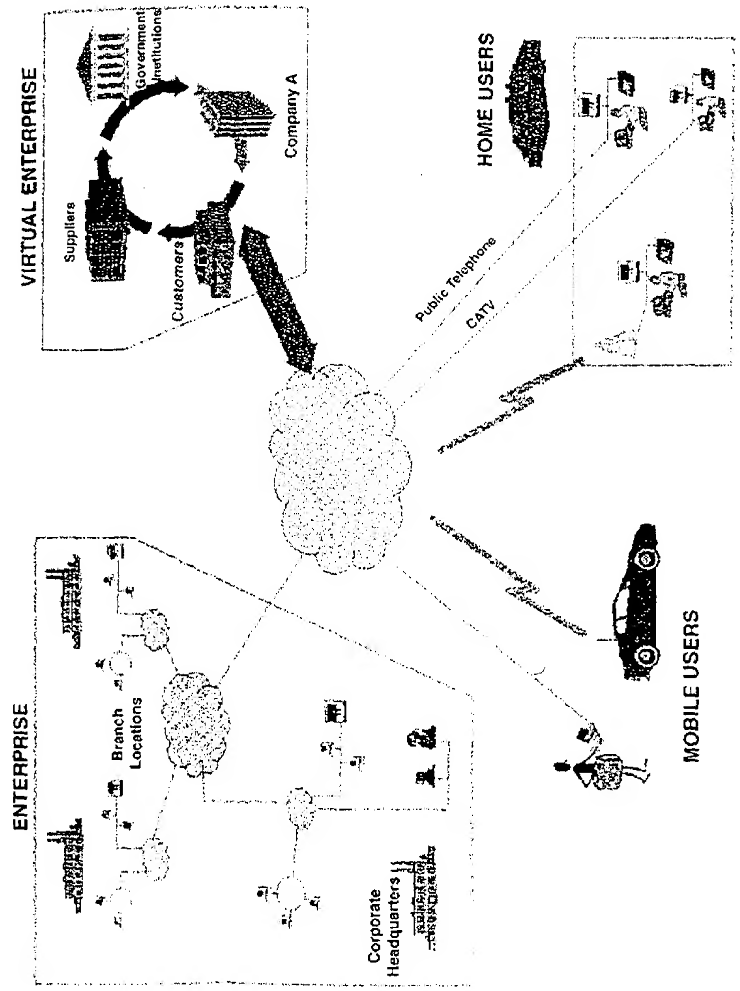

Exhibit 1. Key Network Characteristics.

Mobility

As individuals drive out new requirements, an "anywhere, anytime" computing paradigm must be addressed to support new classes of personal devices. These new devices no longer follow the "bigger, better, faster" characteristics of legacy workstations and servers and hosts but rather the "smaller, cheaper, faster" characteristics of phones, personal digital assistants (PDAs), and laptop computers. This forces the network to support more devices, more variety of devices, and the added overhead of intermittent connectivity.

Distributed Computing

Although most enterprises are still hierarchical, some are flattening their processes by introducing more autonomy. This requires systems to support distributed data, applications, and infrastructure. Because the network is the only part of the infrastructure that has a logical and physical end-to-end view of all resources, the network architecture must provide services to help manage processes that transcend central implementation.

Public Access

Just as enterprise networks are extended to business partners to create virtual enterprises, the enterprise network may also be opened to interaction with the general public (customers, potential customers, etc.). Supporting public access (e.g., internet access, access via public kiosks, etc.) and the resulting virtual communities requires specialized network services relating to security, directories, heterogeneous platforms, etc.

Open Network Services and Interfaces

The communication architecture must support common, open network services and interfaces that are easily shared. Not only must there be well defined standards that can be shared between the client and server, but an "intelligent network" role will need to exist to help proxy capabilities as well. This will allow more rapid expansion of enterprises as they take advantage of virtualization.

The Virtual Enterprise

As more enterprises begin to partner and cooperate, networks will need to support relationships with services that never had to exist before. The challenge to the communication architecture is not in providing the connections but in enabling the end-to-end processes associated with them. Many of these processes will require the network to provide secure, independent, reliable, dynamic services that transcend organizational boundaries.

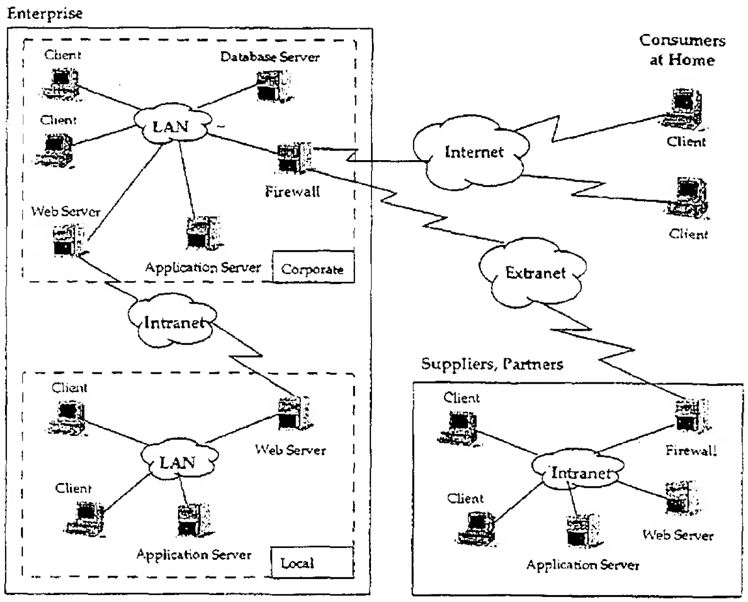

Exhibit 2. Basic Network Components.

THE COMMUNICATIONS ARCHITECTURE

Exhibit 2 is a representation of the physical networking environment of netcentric solutions.

This physical environment is supported by a logical representation of a netcentric execution architecture, discussed in Chapter 3 (Exhibit 3). The components of the architecture that represent the network architecture have been colored gray.

The Communication Services component on the client and server and the communication fabric component represent a high-level view of the communications architecture. The remainder of this chapter focuses on a description of the communication services and the communications fabric portions of the Netcentric Execution Architecture. Exhibit 4 illustrates a further breakdown of the network-specific layers of a netcentric communication architecture.

Each layer (e.g., communication services layer, transport service layer, and the network media layer) contains specific network-related services

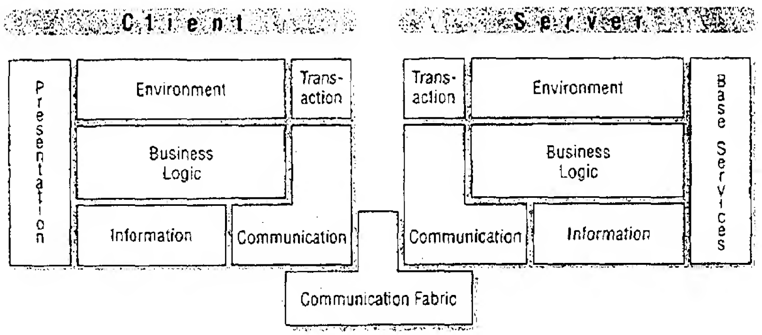

Exhibit 3. Netcentric Execution Architecture.

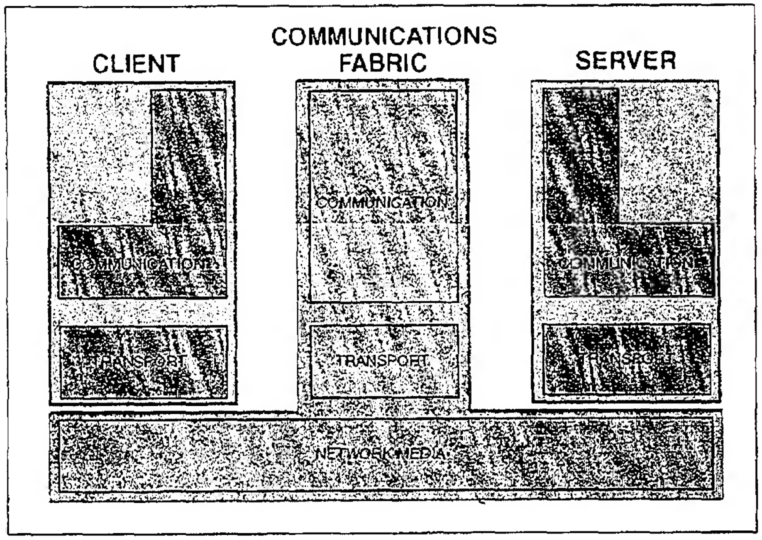

Exhibit 4. Network-Specific Layers of the Communications Architecture.

that are needed to deliver the necessary functionality. To be fully functional, a netcentric architecture requires services from each of the three layers. Within a layer, individual services are selected to deliver the necessary functionality. The services provided in these three layers enable the

applications and higher level services to be isolated from the intricacies of the low-level network (e.g., developing application interfaces directly with complex communications protocols).

Communication Services Layer

The Communication Services layer manages the interaction of distributed processes over the network. This layer enables an application to interact transparently with other applications regardless of whether they reside on the same computer or on a remote computer. The Communication Services layer performs four distinct functions:

- Manages communications between applications
- Initiates and manages the transfer of information between processes over the network
- Provides specialized interface and communication management capabilities based on the type of resource accessed so that network nodes can intelligently interact with distributed resources
- Provides interfacing and translation to ensure that information received is in a readable format for the local system

Transport Services Layer

The Transport Services layer provides capabilities for transferring data through the network to the ultimate destination. Its primary functions include transporting data (including reliability, security, and quality of service) and transporting voice calls.

Network Media Services Layer

The Network Media Services layer performs the low-level transfer of data between network nodes, using physical media such as wiring. Its primary functions include

- Performing low-level transfer of data between network nodes
- Managing low-level signaling across physical media
- Physical wiring, cabling and radio frequency spectrum

Each of these layers plays a distinctive role in the delivery of information from one computing device to another. An analogy might better clarify their distinctive roles. Consider a passenger train moving toward its destination.

The tracks, railroad switches, lights, and stations are performing similar functions as Network Media Services in a Communications Architecture.

The train itself — including engine, cars, and conductor — provides the Transport Services.

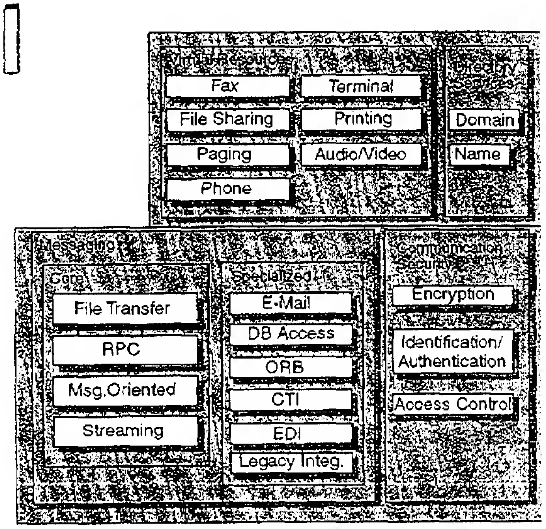

Exhibit 5. Communication Services.

Additionally, at the station, passengers have various services to choose from: the express train, dinner train, destination, and so forth. These services at the station are analogous to the *communication services* component of the architecture.

The rest of this chapter looks at each of these layers in more detail.

COMMUNICATION SERVICES

There are five primary communications services categories (Exhibit 5):

- Core Messaging services
- Specialized Messaging services
- Communications Security services
- Virtual Resource services
- Directory services

CORE MESSAGING SERVICES

Broadly defined, messaging is sending information or commands between two or more recipients. Recipients may be computers, people, or processes in a computer. To send this message, a protocol (or in some cases, multiple protocols) is used that both the sender and receiver can understand. A protocol is a set of rules describing, in technical terms, how two end points should exchange information. Protocols exist at several levels during the exchange of information. Protocols facilitate transport of the message carrying the information. Both end points must recognize and observe the protocol. As an example, a common protocol in today's networks is the Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP is the principle method for transmitting data over the Internet today. This protocol is responsible for ensuring that a series of data packets sent over a network arrives at the destination and is properly sequenced.

Messaging services transfer formatted information from one process to another. By drawing upon messaging services, applications can shield themselves from the complexity of the low-level Transport services. There are three key messaging styles used to support Interprocess Communication (IPC): Store and Forward and Synchronous and Asynchronous Messaging.

Store and forward messaging provides deferred message processing. For example, store and forward messaging may use an e-mail infrastructure upon which to build applications. Common uses would be for forms routing and e-mail.

Synchronous messaging allows an application to send a message to another application and wait for a reply before continuing. Synchronous messaging is typically used for update and general business transactions. It requires time-out processing to allow the application to reacquire control in the event of failure.

Asynchronous messaging allows an application to send a message to another application and continue processing before a reply is received. Asynchronous messaging is typically used for larger retrieval type processing, such as retrieval of larger lists of data than can be contained in one message.

Messaging styles are important because they serve as the primary link to the application and business requirements. For example, suppose a business process requiring a series of processing steps needs to be automated. Additionally, each step needs to be performed in sequence at real time. Before continuing to the next step of the process, an application must know if the previous step was successful. Because of the send, receive, continue nature of the business process, the more appropriate messaging style for this application is synchronous messaging.

In addition to the messaging styles, interprocess messaging is typically implemented in one of two ways:

- *Function based:* uses the subroutine model of programming. The message interface is built upon the calling program passing the appropriate parameters and receiving the returned information.

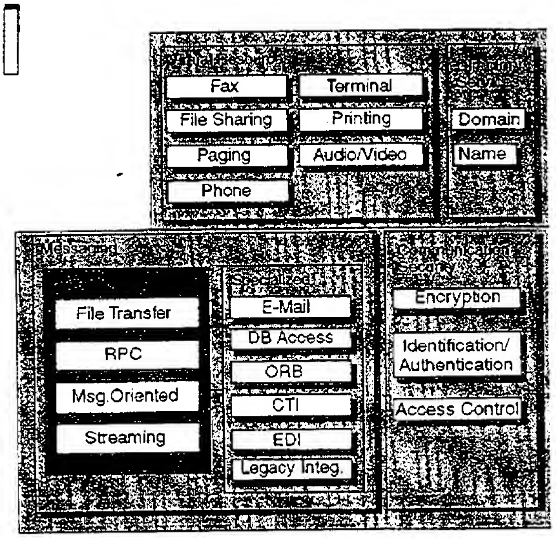

Exhibit 6. Core Messaging Services.

- *Message based:* uses a defined message format to exchange information between processes. While a portion of the message may be unstructured, a defined header component is normally included. A message-based approach is not limited to the call/return structure of the function-based model and can be used in a conversational manner.

Core messaging services can be divided into the following services (Exhibit 6):

- File transfer services
- RPC (Remote procedure call) services
- Message-Oriented services
- Streaming services

File Transfer Services

File Transfer services enable the copying and receiving of files or other large blocks of data between two resources. Exhibit 7 depicts File Transfer, in which a bulk data transfer occurs (possibly in either direction). Note that a file transfer copies a file, resulting in a copy on both machines.

The following are examples of File Transfer protocols and standards.

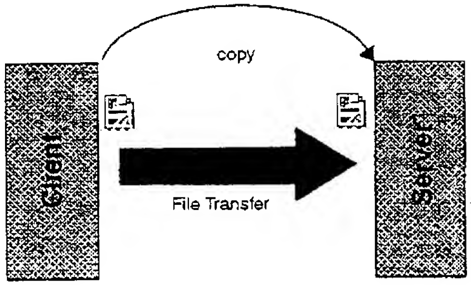

Exhibit 7. File Transfer.

File Transfer Protocol (FTP). Allows users to upload and download files across the network. FTP also provides a mechanism to obtain file name, directory name, attributes, and file size information. Remote file access protocols such as Network File System (NFS) also use a block transfer method but are optimized for on-line read/write paging of a file.

Hyper-Text Transfer Protocol (HTTP). Within a Web-based environment, Web servers transfer HTML pages to clients using HTTP. HTTP can be thought of as a lightweight file transfer protocol optimized for transferring small files. HTTP reduces the inefficiencies of the FTP protocol. HTTP runs on top of TCP/IP and was developed specifically for the transmission of hypertext between client and server.

Secure Hypertext Transfer Protocol (S-HTTP). A secure form of HTTP, mostly for financial transactions on the Web. S-HTTP has gained a small level of acceptance among merchants selling products on the Internet as a way to conduct financial transactions (using credit card numbers or passing sensitive information) without the risk of unauthorized people intercepting this information. S-HTTP incorporates various cryptographic message formats such as DSA and RSA standards into both the Web client and the Web server.

File Transfer and Access Management (FTAM). The OSI (Open Systems Interconnection) standard is used for file transfer, file access, and file management across platforms.

Remote Procedure Calls (RPC) Services

RPCs are a type of protocol by which an application sends a request to a remote system to execute a designated procedure using the supplied arguments and return the result.

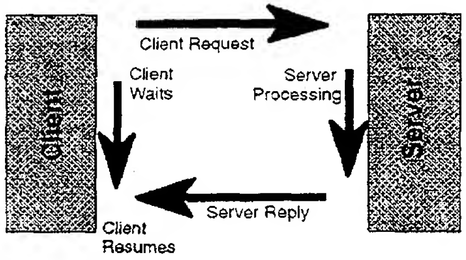

Exhibit 8. RPC Messaging.

RPCs emulate the *function call mechanisms* found in procedural languages (e.g., the C language). This means that control is passed from the main logic of a program to the called function, with control returning to the main program once the called function completes its task. Because RPCs perform this mechanism across the network, they pass some element of control from one process to another, for example, from the client to the server. Because the client is dependent on the response from the server, it is normally blocked from performing any additional processing until a response is received. This type of synchronous data exchange is also referred to as blocking communications.

Exhibit 8 depicts RPC messaging, in which the message originator stops processing while waiting for a reply.

Message-Oriented Services

Message-Oriented Services refers to the process of distributing data and control through the exchange of records known as "messages." Message-Oriented Services provide the application developer with a set of simple verbs (e.g., connect, send, receive, and disconnect) that are used to exchange information with other distributed applications.

For example, to send data to a remote process, the application developer uses a send verb. This verb along with the appropriate parameters (e.g., data to be sent and the process's logical name) are included as part of the application code.

Once the verb is called, the Message-Oriented Services are responsible for managing the interface to the underlying communications architecture via the communications protocol APIs and ensuring the delivery of the information to the remote process. This interface may require that Message-Oriented Services have the following capabilities:

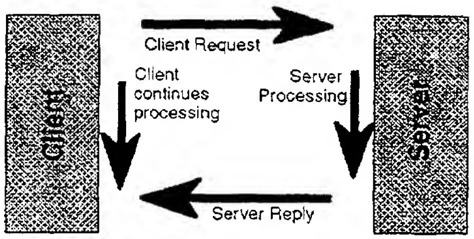

Exhibit 9. Message Passing.

- Translating mnemonic or logical process names to operating system compatible format
- Opening a communications session and negotiating parameters for the session
- Translating data to the proper format
- Transferring data and control messages during the session
- Recovering any information if errors occur during transmission
- Passing results information and status to the application

An application continues processing after executing a Message-Oriented Services verb, allowing the reply to arrive at a subsequent time. Thus, unlike RPCs, Message-Oriented Services implements a "nonblocking" messaging architecture.

Message-Oriented Services products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes).

There are three types of Message-Oriented Services commonly implemented:

- Message Passing
- Message Queuing
- Publish and Subscribe

Message Passing. This is a direct, application-to-application communication model. An application request is sent in the form of a message from one application to another. The communication method can be either synchronous (in this case the sending applications waits for a response back from the receiving application, like RPCs) or asynchronous (through callback routines). In a message-passing model, a direct link between two applications that participate in the message exchange is always maintained (Exhibit 9).

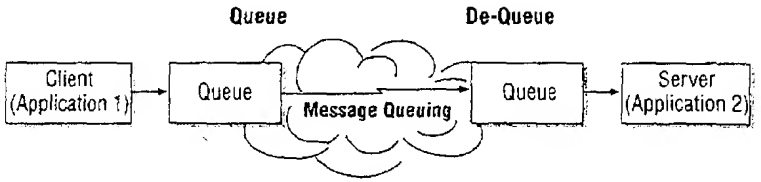

Exhibit 10. Message Queuing.

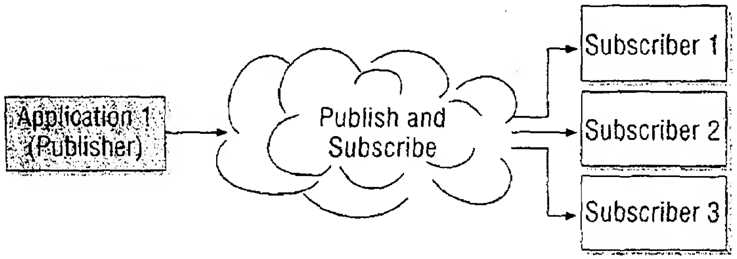

Exhibit 11. Publish and Subscribe Messaging.

Message Queuing. Message Queuing (also known as Store and Forward) is an indirect application to application communication model that allows applications to communicate via message queues rather than by calling each other directly (Exhibit 10). Message queuing is asynchronous by nature and connectionless, meaning that the recipient need not be directly available when the message is sent. Moreover, it implies support for reliable, guaranteed, and assured (nonduplicate) message delivery.

Publish and Subscribe. Publish and Subscribe (also known as Push messaging) is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events (Exhibit 11). An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it.

Streaming Services

Streaming is the process of transferring time-sensitive data streams (e.g., video and/or audio) in real time. Streaming differs from the other types of Core Messaging services in that it delivers a continuous, one-way stream of data rather than the relatively short messages associated with RPC and Message-Oriented messaging or the large, batch transfers associated with File Transfer. (While the media stream is one-way from the server to the client, the client can issue stream controls to the server.) Streaming may be used to deliver video, audio, and/or other real-time content across the Internet or within enterprise networks.

Functionality	Sample Protocol options	Architecture Service
Controlling media delivery	RTSP or proprietary	Streaming messaging service
Monitoring data stream	RTCP or proprietary	Streaming messaging service
End-to-end delivery of stream	RTP or proprietary	Streaming messaging service
Message transport	UDP, Multicast UDP, TCP	Message transport service
Packet forwarding/Internetworking	IP, IP multicast	Packet forwarding/Internetworking service

Exhibit 12. Streaming Architecture Options

Streaming is an emerging technology. While some multimedia products use proprietary streaming mechanisms, other products incorporate standards. Data streams are delivered using several protocols that are layered to assemble the necessary functionality. The following are examples of emerging standards for streaming protocols.

Real-Time Streaming Protocol (RTSP). RTSP is the proposed Internet protocol for establishing and controlling on-demand delivery of real-time data. For example, clients can use RTSP to request specific media from a media server, to issue commands such as play, record and pause, and to control media delivery speed. Because RTSP simply controls media delivery, it is layered on top of other protocols, such as the following.

Real-Time Transport Protocol (RTP). Actual delivery of streaming data occurs through real-time protocols such as RTP. RTP provides end-to-end data delivery for applications transmitting real-time data over multicast or unicast network services. RTP conveys encoding, timing, and sequencing information to allow receivers to properly reconstruct the media stream. RTP is independent of the underlying transport service, but it is typically used with UDP. It may also be used with Multicast UDP, TCP/IP, or IP Multicast.

Real-Time Control Protocol (RTCP). RTP is augmented by the RTCP. RTCP allows nodes to identify stream participants and communicate about the quality of data delivery.

Exhibit 12 summarizes the protocol layering that supports Streaming.

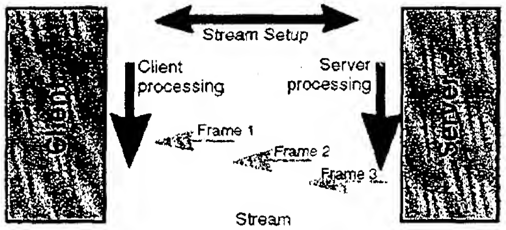

Exhibit 13. Streaming.

A key attribute of any streaming architecture is the adherence to a flow of time-sequenced data packets. Each series of packets contains the necessary information to play the next segment in a sound or video clip. Exhibit 13 highlights the one-way, time-sequenced nature of the flow of data packets for a Streaming architecture.

SPECIALIZED MESSAGING SERVICES

Specialized Messaging services extend the Core Messaging services to provide additional functionality. Specialized Messaging services may extend Core Messaging services in the following general ways:

- Providing messaging among specialized systems by drawing upon basic messaging capabilities
- Defining specialized message layouts
- Defining specialized intersystem protocols
- Suggesting ways in which messaging draws upon directory and security services to deliver a complete messaging environment

An example of a specialized messaging service is E-Mail Messaging. E-Mail Messaging is an implementation of a store-and-forward Message-Oriented Services, in that E-Mail Messaging defines specialized, mail-related message layouts and protocols that utilize store-and-forward messaging.

Specialized Messaging services is comprised of the following categories (Exhibit 14):

- E-Mail Messaging
- CTI Messaging
- EDI Messaging
- Object Request Broker Messaging
- Database Access Messaging
- Legacy Integration Messaging

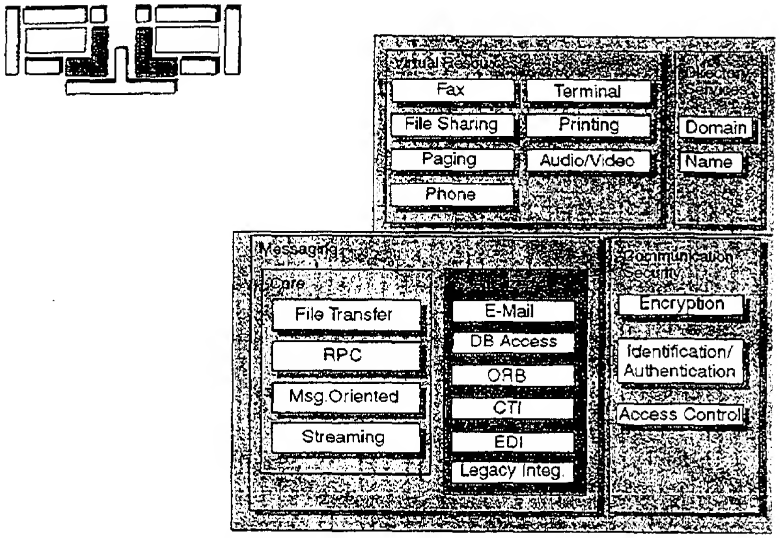

Exhibit 14. Specialized Messaging Services.

E-Mail Messaging Services

E-Mail Messaging services reliably exchange messages using the store-and-forward messaging style. E-Mail message systems traditionally include a rudimentary form of directory services (discussed later). While some e-mail products use proprietary protocols, the following are examples of E-Mail-related standards:

X.400. The X.400 message handling system standard defines a platform independent standard for store-and-forward message transfers among mail servers. X.400 is often used as a backbone E-Mail service, with gateways providing interconnection with end-user systems.

Simple Mail Transfer Protocol (SMTP). SMTP is a UNIX/Internet standard for transferring E-Mail among servers.

Multi-Purpose Internet Mail Extensions (MIME). MIME is a protocol that enables Internet users to exchange multimedia E-Mail messages.

Post Office Protocol (POP). POP3 is used to distribute E-Mail from an SMTP server to the actual recipient.

Internet Message Access Protocol, Version 4 (IMAP4). IMAP4 allows a client to access and manipulate E-Mail messages on a server. IMAP4 permits manipulation of remote message folders, called "mailboxes," in a way that is functionally equivalent to local mailboxes. IMAP4 also provides the capability for an off-line client to resynchronize with the server. IMAP4 includes standards for message handling features that allow users to download message header information and then decide which E-Mail message contents to download.

Database Messaging Services

Database Messaging services (also known as Database Access Middleware, or DBAM) provide connectivity for clients to access databases throughout the enterprise. Database messaging software draws upon basic interprocess messaging capabilities (e.g., RPCs) to support database connectivity. DBAM can be grouped into one of three categories:

- Open
- Native
- Gateway

Open database messaging services typically provide single applications seamless access to multiple data sources, both relational and nonrelational, through a standard application programming interface (API) set. Examples include ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity). ODBC is considered an industry de facto standard.

By contrast, *native* database messaging services are those services, usually proprietary, provided by the DBMS vendor. Examples include SQL*Net for Oracle DBMS and DB-LIB for Sybase DBMS.

Additionally, *gateway* database messaging services can be used to facilitate migration of data from one environment to another. For example, if data in a DB2 environment needs to be integrated with data in a Sybase environment, Gateway DBAM can enable the integration.

Object Request Broker (ORB) Messaging Services

ORB Messaging enables objects to transparently make requests of, and receive responses from, other objects located locally or remotely. Objects communicate through an ORB. An ORB enables client objects to access server objects either locally or remotely over a network and invoke operations (i.e., functions and methods) on them. ORBs typically provide interoperability between heterogeneous client and server environments across languages and/or operating systems and/or network protocols. In that respect, some have said that ORBs will become a kind of "ultimate middleware" for truly distributed processing. A standardized Interface Definition Language (IDL) defines the interfaces that applications must use to access the ORB Services. The two major Object Request Broker standards/implementations are

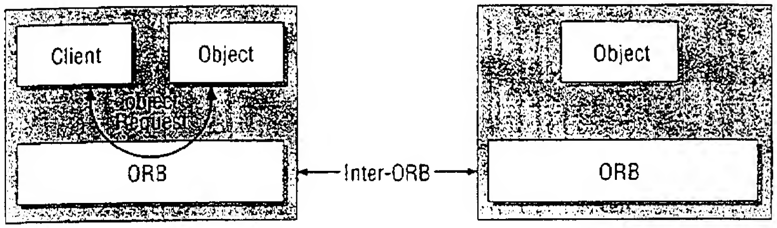

Exhibit 15. CORBA-Based Object Request Broker Messaging.

- Object Management Group's Common Object Request Broker Architecture (CORBA) (www.omg.org)
- Microsoft's (Distributed) Component Object Model (COM/DCOM) (www.microsoft.com)

CORBA. Common Object Request Broker Architecture (CORBA) is a standard for distributed objects being developed by the Object Management Group (OMG). The OMG is a consortium of software vendors and end users. Many OMG member companies are developing commercial products that support the CORBA standards and/or are developing software that use these standards. CORBA provides the mechanism by which objects transparently make requests and receive responses, as defined by OMG's ORB. The CORBA ORB is an application framework that provides interoperability between objects, built in different languages, running on different machines in heterogeneous distributed environments.

The OMG's Internet Inter-Orb Protocol (IIOP) specifies a set of message formats and common data representations for communication between ORBs over TCP/IP networks. CORBA-based Object Messaging is summarized in Exhibit 15.

Component Object Model. Component Object Model (COM) is a client/server object-based model, developed by Microsoft, designed to allow software components and applications to interact with each other in a uniform and standard way. The COM standard is partly a specification and partly an implementation. The specification defines mechanisms for creation of objects and communication between objects. This part of the specification is paper based and is not dependent on any particular language or operating system. Any language can be used as long as the standard is incorporated. The implementation part is the COM library that provides a

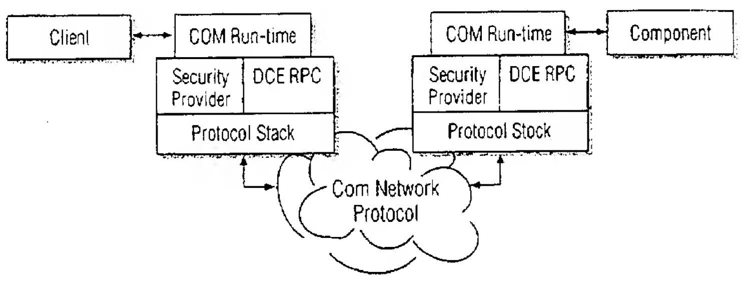

Exhibit 16. Microsoft COM/DCOM Messaging.

number of services that support a mechanism that allows applications to connect to each other as software objects (Exhibit 16).

COM is not a software layer through which all communications between objects occur. Instead, COM serves as a broker and name space keeper to connect a client and an object, but, once that connection is established, the client and object communicate directly without having the overhead of passing through a central piece of API code. Originally conceived of as a compound document architecture, COM has been evolved to a full object request broker including recently added features for distributed object computing. DCOM (Distributed COM) contains features for extending the object model across the network using the DCE RPC mechanism. In sum, COM defines how components should be built and how they should interact. DCOM defines how they should be distributed. Currently COM/DCOM is only supported on Windows-based machines. However, third-party vendors are in progress of porting this object model to other platforms such as Macintosh, UNIX, etc.

CTI Messaging Services

Computer-Telephone Integration (CTI) integrates computer systems and telephone systems to coordinate data and telephony activities. For example, CTI can be used to associate a customer's database entry with the customer's telephone call and route the call accordingly.

CTI Messaging supports communication among clients, CTI servers, PBXs/ACDs, hybrid platforms, networks, and external telephony devices. CTI Messaging relies upon proprietary PBX/ACD APIs, CTI vendor-specific APIs or message sets, and industry-standard APIs.

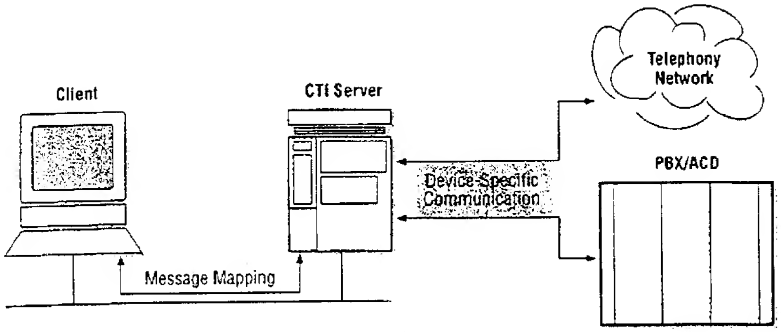

Exhibit 17. CTI Messaging.

CTI Messaging has two primary functions (Exhibit 17):

1. Device-specific communication
   Manages direct communications between telephony devices and data devices.
   Allows applications to control PBXs, key telephone systems, ISDN, analog PSTN, cellular, Centrex, etc. and supports features such as address translation, call setup, call answering, call dropping, and caller ID.
   Provides interface to carrier networks for call delivery and call-related messaging.
2. Message mapping
   Translates device-specific communication to generic API and/or message set

CTI products can be divided into the following categories.

CTI Platform-Specific Products. These can only be implemented on the hardware of a specific vendor.

CTI Telephony-Based API Products. These include proprietary PBX/ACD-based messaging sets, which permit external devices to interface with the vendor's PBX/ACD call and station control logic.

CTI Server/Workstation-Based or Host-Based API Products. These operate on a particular computer vendor's hardware platform and provide call control and messaging functionality.

CTI Cross-Platform Vendors. These products have been ported to multiple hardware platforms/operating systems.

CTI Enabling Solutions. These focus solely on call control and call/application synchronization functions.

CTI Enterprise Solutions. These provide all CTI business functions to varying degrees.

EDI Messaging Services

EDI (Electronic Data Interchange) supports system-to-system messaging among business partners by defining standard message layouts. Companies typically use EDI to streamline commercial transactions in their supply chains.

EDI standards (e.g., EDIFACT, ANSI X12) define record layouts for transactions such as "purchase orders." EDI services include the generation and translation of EDI messages according to the various public message layout standards.

EDI messaging can be implemented via electronic mail or customized message-oriented architectures.

Legacy Integration Services

Legacy services provide gateways to mainframe legacy systems. Design techniques for integration with existing systems can be grouped into two broad categories:

- Front end access: access of information through screens/windows (this will be further discussed in the Terminal Emulation section in Virtual Resources later in this chapter).
- Back end access: this approach tends to be used when existing data stores have information that is needed in the client/server environment, but accessing the information through existing screens or functions is not feasible. Legacy messaging services typically include remote data access through gateways. A database gateway provides an interface between the client/server environment and the legacy system. The gateway provides an ability to access and manipulate the data in the legacy system.

COMMUNICATION SECURITY SERVICES

As organizations open up their computing resources to business partners, customers, and a broader audiences of employees, security becomes one of the hottest topics in most discussions. This section focuses on network communications-related security. For a broader perspective on security in netcentric environments, refer to Chapter 28. This chapter will introduce some of the key communications architecture security concepts.

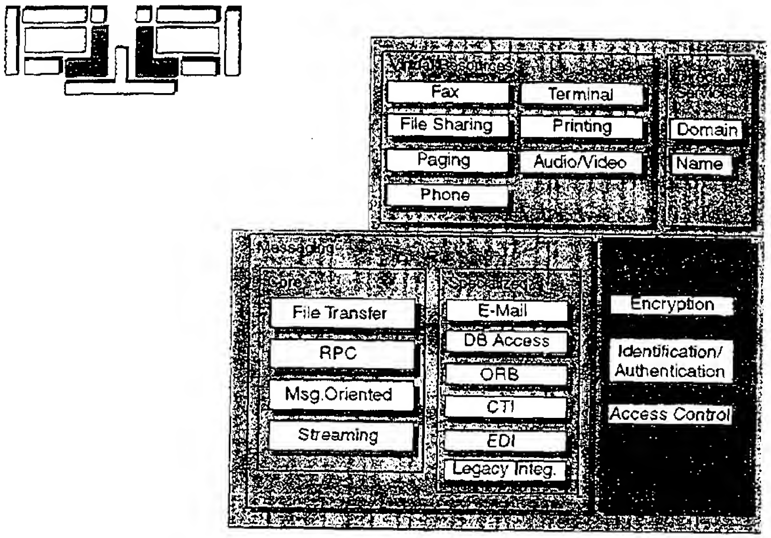

Exhibit 18. Communication Security Services.

Communications security services can be broken down into the following three categories (Exhibit 18):

- Encryption Services
- Identification and Authentication Services
- Access Control Services

Encryption Services

Encryption services encrypt data prior to network transfer to prevent unauthorized interception. (Note that encryption can occur within the Communication Services layer, the Transport Services layer, or the Network Media Services layer.) Within the Communication Services layer, encryption occurs at the top of the protocol stack and is typically performed in an application (e.g., in an e-mail application). This is an end-to-end approach that can leave the remainder of the protocol stack (i.e., the Transport services and the Network Media services) unaffected. Refer to Transport Security topic in the Transport Services section for more information on security.

Identification/Authentication Services

Identification/Authentication services verify network access requests by validating that users are who they claim to be. For secure systems, one or more Identification/Authentication mechanisms can be used to validate authorized users and integrated with Access Control Services to verify which functions and data they have access to. Within the corporate network, Identification/Authentication services are often included in directory services products like Novell's NDS (NetWare Directory Services) or Microsoft's Windows NT Domain Services. These products require the user to have an established account and supply a password before access is granted to resources through the directory.

Identification/Authentication for accessing resources across an Internet or intranet is not as simple and is a rapidly evolving area. Web sites need to restrict access to areas of information and functionality to known customers or business partners. More granular Identification/Authentication services are required where sensitive individual customer account information must be protected from other customers.

Identification/Authentication can occur through various means.

Basic ID/Authentication. This requires that the Web client supply a user name and password before servicing a request. Basic ID/Authentication does not encrypt the password in any way, and thus the password travels in the clear over the network where it could be detected with a network sniffer program or device. Basic ID/Authentication is not secure enough for banking applications or anywhere where there may be a financial incentive for someone to steal someone's account information. It is, however, the easiest mechanism to set up and administer and requires no special software at the Web client.

ID/Password Encryption. This offers a somewhat higher level of security by requiring that the user name and password be encrypted during transit. The user name and password are transmitted as a scrambled message as part of each request because there is no persistent connection open between the Web client and the Web server.

Digital Certificates or Signatures. These are encrypted digital keys that are issued by a third party "trusted" organization (i.e., Verisign). They are used to verify a user's authenticity.

Hardware Tokens. These are small physical devices that may generate a one-time password or that may be inserted into a card reader for ID/Authentication purposes.

Virtual Tokens. These are typically a file on a floppy or hard drive used for ID/Authentication (e.g., Lotus Notes ID file).

Biometric Identification. This involves the analysis of biological characteristics (such as fingerprints, voice recognition, or retinal scans) to verify an individual's identify.

Access Control Services

When a user requests access to network resources, the Access Control service determines if the user has the appropriate permissions or privileges and either allows or disallows the access. (This occurs *after* the user has been properly identified and authenticated.)

The following are examples of ways to implement Access Control services.

Network Operating Systems. Access Control services are bundled with all network operating systems to control user access to network resources.

Application Proxies. An application-level proxy, or application-level gateway, is a robust type of firewall. (A firewall is a system that enforces an access control policy between a trusted internal network and an untrusted external network.) The application proxy acts at the application level rather than the network level. The proxy acts as a go-between for the end user by completing the user-requested tasks on its own and then transferring the information to the user. The proxy manages a database of allowed user actions, which it checks prior to performing the request.

Filters. World Wide Web filters can prevent users from accessing specified content or Internet addresses. Products can limit access based on keywords, network addresses, time-of-day, user categories, etc. Filters are typically implemented on a firewall.

Servers, Applications, and Databases. Access Control can occur locally on a server to limit access to specific system resources or files. Applications and databases can also authorize users for specific levels of access within their control. (This functionality is within the Environment Services grouping in the execution architecture.)

DIRECTORY SERVICES

Directory services will play a major role in the future of netcentric computing, primarily because of the increasingly distributed and dynamic nature of netcentric environments. Directory services manage information about resources on the network and perform a variety of processes. These processes range from simple name-to-address resolution (e.g., when *www.ac.com* is typed in a browser connected to the Internet, that name resolves to IP address 204.167.146.195.) to the logical integration of heterogeneous systems to create a common view of resources.

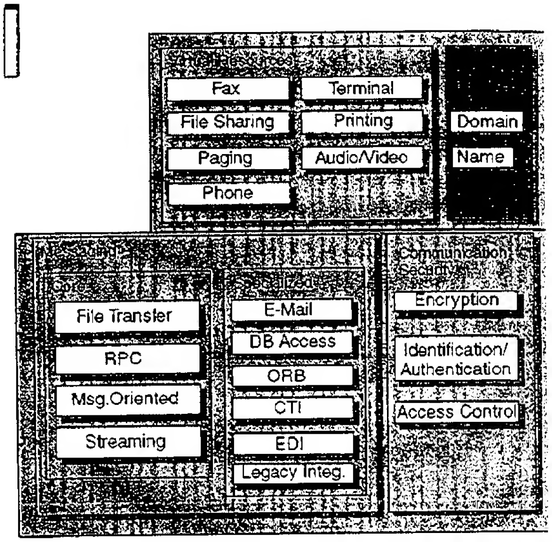

Exhibit 19. Directory Services.

Directory Services typically perform one or many of the following functions

- Store information about network resources and users, and track relationships
- Organize resource access information in order to aid in the location of and access to other resources throughout the network
- Provide location transparency, because resources are accessed through a directory rather than based on their physical location
- Convert between logical resource names and physical resource addresses
- Interact with Security services such as identification/authentication and access control services to maintain necessary access permissions and privileges
- Provide single network logon to file and print resources; in certain cases, provide single network logon for network applications integrated with the directory services
- Distribute and synchronize directory information throughout the environment (for reliability and location-independent access)

Directory Services is comprised of two subservices: Name Services and Domain Services (Exhibit 19).

Name Services

The Name service creates a logical "pronounceable" name in place of a binary machine number. These services could be used by other communications services such as File Transfer, Message Services, and Terminal Services. A Name service can be implemented on its own or as part of a full-featured Directory service.

Domain Services

A network domain is a set of network nodes under common control (i.e., common security and logins, unified addressing, coordinated management, etc.). Domain services manage these types of activities for the network nodes in a domain. Domain services may be limited in their ability to support heterogeneous systems and in the ability to scale to support the enterprise.

Most Directory services running today tend either to provide limited functionality or to be highly proprietary. In fact, many organizations maintain multiple directories from e-mail to printer and host information. In a netcentric environment, it is crucial to provide seamless location of, and access to, resources, individuals, and applications. Emerging directory service technologies such as the Lightweight Directory Access Protocol (LDAP) may prove key in providing integrated, open Directory services for netcentric applications.

VIRTUAL RESOURCE SERVICES

Virtual Resource Services proxy or mimic the capabilities of specialized, network-connected resources. This allows a generic network node to emulate a specialized physical device. In this way, network users can interface with a variety of specialized resources.

A common example of a Virtual Resource service is the capability to print to a network printer as if it were directly attached to a workstation.

Virtual Resource Services include the following (Exhibit 20):

- Terminal Services
- Print Services
- File Sharing Services
- Phone Services
- Fax Services
- Audio/Video Services
- Paging Services

Terminal Services

Terminal Services allow a client to connect to a nonlocal host via a network and to emulate the profile (e.g., the keyboard and screen characteristics)

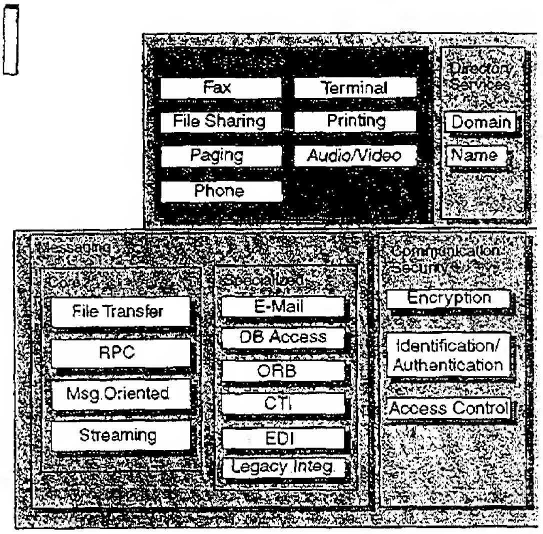

Exhibit 20. Virtual Resource Services.

required by the host application. For example, when a workstation application logs on to a mainframe, the workstation functions as a dumb terminal. Terminal Services receive user input and send data streams back to the host processor. If connecting from a PC to another PC, the workstation might act as a remote control terminal (e.g., PC Anywhere).

The following are examples of Terminal services.

Telnet: a simple and widely used terminal emulation protocol that is part of the TCP/IP communications protocol. Telnet operates establishing a TCP connection with the remotely located login server, minicomputer, or mainframe. The client's keyboard strokes are sent to the remote machine while the remote machine sends back the characters displayed on the local terminal screen.

3270 emulation: emulation of the 3270 protocol that is used by IBM mainframe terminals.

tn3270: a Telnet program that includes the 3270 protocol for logging onto IBM mainframes; part of the TCP/IP protocol suite.

X Window System: allows users to simultaneously access applications on one or more UNIX servers and display results in multiple windows on a local display. Recent enhancements to XWS include integration with the Web and optimization of network traffic (caching, compression, etc.).

Remote control: while terminal emulation is typically used in host-based environments, remote control is a sophisticated type of client/server Terminal service. Remote control allows a client computer to control the processing on a remote desktop computer. The GUI on the client computer looks as if it is the GUI on the remote desktop. This makes it appear as if the remote applications are running on the client.

rlogin: a remote terminal service implemented under BSD UNIX. The concept behind rlogin is that it supports "trusted" hosts. This is accomplished by having a set of machines that share common file access rights and logins. The user controls access by authorizing remote login based on a remote host and remote user name. This service is generally considered a security risk and avoided in most business system configurations.

Print Services

Print services connect network workstations to shared printers. The administration of Print Services is usually handled by a print server. Depending on the size of the network and the amount of resources the server must manage, the print server may run on a dedicated machine or on a machine that performs other server functions. Print servers queue print jobs sent to network printers, which are stored in the server's print buffer and then sent to the appropriate network printer as it becomes available. Print services can also provide the client with information, including print job status, and can manage in-progress print jobs.

File Sharing Services

File Sharing Services allow users to view, manage, read, and write files that may be located on a variety of platforms in a variety of locations. File Sharing services enable a unified view of independent file systems. This is represented in Exhibit 21, which shows how a client can perceive remote files as being local.

File Sharing services typically provide some or all of the following capabilities:

Transparent access: access to remote files as if they were local.

Multiuser access: distribution and synchronization of files among multiple users, including file locking to manage access requests by multiple users.

File access control: use of Security services (user authentication and authorization) to manage file system security.

Multiplatform access: access to files located on various platforms (e.g., UNIX, NT, etc.).

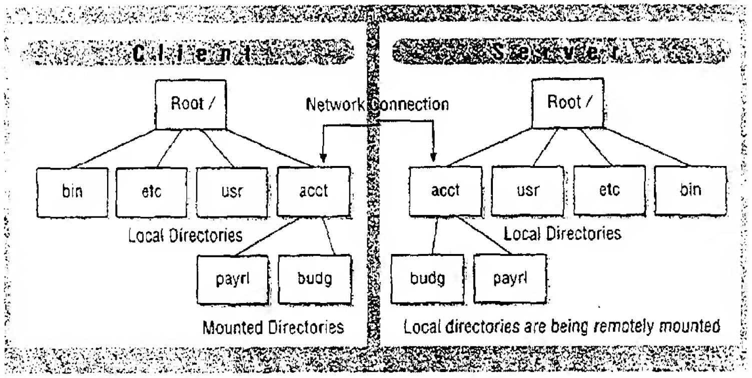

Exhibit 21. UNIX File Sharing Services Example.

Integrated file directory: a logical directory structure that combines all accessible file directories, regardless of the physical directory structure.

Fault tolerance: use of primary and replica file servers to ensure high availability of file system.

Scalability: ability to integrate networks and distributed file systems of various sizes

Phone Services

Phone virtual resource services extend telephony capabilities to computer platforms. For example, an application on a desktop computer can place and receive telephone calls for the user. Phone virtual resource services may be used in customer care centers, help desks, or any other environment in which it is useful for a computer to replace a telephone handset.

Phone services enable clients, servers, and specialized telephony nodes (PBXs, ACDs, etc.) to control the telephony environment through the following telephony controls:

- Call control
- Controls telephone features
- Controls recorded messages
- Manipulates real time call activities (e.g., make call, answer, transfer, hold, conference, mute transfer, release, route call, call treatments, and digits collected)
- Telephone status control
- Controls telephone status functions

- Logs users in and out of the system
- Sets ready, not ready, and make busy statuses for users

The following are examples of uses of Phone virtual resources.

PC Telephony. PC telephony products allow desktop computers to act as conduits for voice telephone calls.

Internet Telephony. Internet telephony products enable voice telephone calls (and faxing, voice mail retrieval, etc.) through the Internet. For example, an Internet telephony product can accept voice input into a workstation, translate it into an IP data stream, and route it through the Internet to a destination workstation, where the data is translated back into audio.

Desktop Voice Mail. Various products enable users to manage voice mail messages using a desktop computer.

Fax Services

Fax Services provide for the management of both inbound and outbound fax transmissions. If fax is used as a medium for communicating with customers or remote employees, inbound fax services may be required for centrally receiving and electronically routing faxes to the intended recipient. Outbound fax services can be as simple as supporting the sharing on the network of a single fax machine or group of machines for sending faxes.

Examples of Fax service functionality include the following:

- Managing incoming faxes
- Receiving faxes via the telephone network
- Queuing faxes
- Routing and distributing faxes
- Displaying or printing faxes
- Managing outgoing faxes
- Generating faxes
- Queuing faxes
- Transferring faxes via the telephone network

Fax services can provide centrally managed faxing capabilities, thus eliminating the need for fax modems on every workstation. A fax server generally provides fax services to clients, such as receiving, queuing, and distributing incoming faxes and queuing and sending outgoing faxes. Clients can view faxes and generate faxes to be sent.

Applications may compose and transfer faxes as part of notifying users or delivering information. For example, an application may use Fax services to add customer-specific information to a delivery receipt form and fax the form to a customer.

Audio/Video Services

Audio/Video services allow nodes to interact with multimedia data streams. These services may be implemented as audio only, video only, or combined audio/video.

Audio Services. Audio services allow components to interface with audio streams such as the delivery of music or radio content over data networks.

Video Services. Video services allow components to interface with video streams such as video surveillance. Video services can add simple video monitor capabilities to a computer, or they can transform the computer into a sophisticated video platform with the ability to generate and manipulate video.

Combined Audio/Video Services. Video and audio content is often delivered simultaneously. This may be accomplished by transferring separate audio and video streams or by transferring a single interleaved stream. Examples include video conferencing and television (traditional or interactive).

Audio/Video services can include the following functionality:

- Streaming content (audio, video, or both) to end users
- Managing buffering of data stream to ensure uninterrupted viewing/listening
- Performing compression and decompression of data
- Managing communications protocols to ensure smooth delivery of content
- Managing library of stored content and/or manages generation of live content

Audio/Video services draw upon lower-level services such as streaming (see Streaming Messaging services) and IP Multicast (see Packet Forwarding/Internetworking services) to efficiently deliver content across the network.

Paging Services

Wireless short messaging (i.e., paging) can be implemented through wireless systems such as paging networks, GSM voice/data networks, PCS voice/data networks, and dedicated wireless data networks.

Paging virtual resource services provide the message formatting and display functionality that allows network nodes to interface with wireless paging systems. This service emulates the capabilities of one-way and two-way pagers (Exhibit 22).

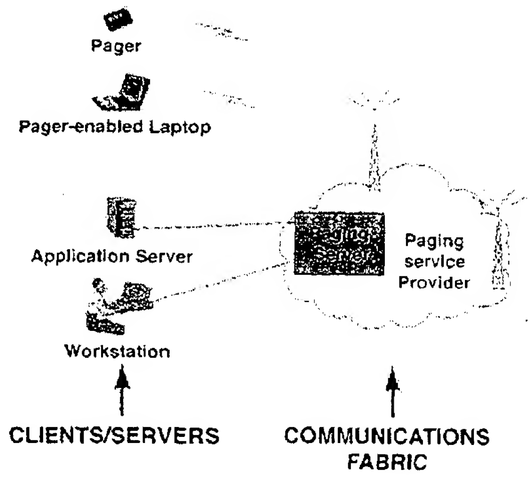

Exhibit 22. Use of a Paging Virtual Resource.

Paging systems allow pages to be generated in various ways:

- E-Mail messages to a specified mailbox
- DTMF (touch tone) signaling to a voice response system
- Encoded digital messages transferred into a paging provider gateway
- Messages transferred to a locally attached two-way wireless pager

COMMUNICATION SERVICES LAYER SUMMARY

Overall, the communication services layer provides the foundation for netcentric applications enabling client/server and virtual resource communications. Selecting the appropriate Communication Services, services that meet the business and applications requirements, is a key step to ensuring a successful Communications Architecture. In addition, ensuring the required Transport Services support the selected Communication Services is important. Transport Services are the subject of the next section.

TRANSPORT SERVICES

Transport Services are the portion of the Communications Architecture that provides the movement of information across a network. While the Communications Fabric includes all the hardware, software, and services between the client and server nodes, the Transport Services play a key role

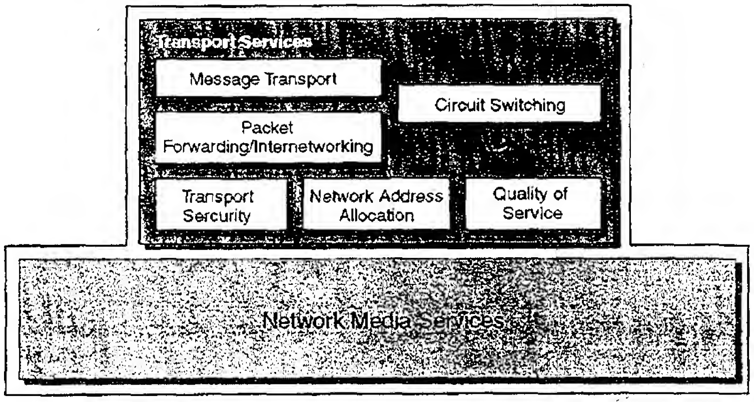

Exhibit 23. Transport Services.

performing network functions across the enterprise or between enterprises. Transport Services include the following (see also Exhibit 23):

- Message Transport services
- Packet Forwarding/Internetworking services
- Circuit Switching services
- Transport Security services
- Network Address Allocation services
- Quality of Service services

Message Transport Services

Message Transport Service are responsible for the end-to-end delivery of messages. They can include the following functionality.

End-to-End Data Transfer. The Message Transport Service formats messages for sending and confirms the integrity of received messages.

Connection Control. The Message Transport service may establish end-to-end (client–server) connections and track addresses and other associated information for the connection. The service also tears down connections and handles hard connection failures.

Reliable Transfer. The Message Transport service may manage reliable delivery of messages through the use of acknowledgments and retransmissions.

Flow Control. The Message Transport service may allow the receiver to govern the rate at which the sender transfers data.

Multiplexing. The Message Transport service may define multiple addresses or ports within a single network node, allowing multiple processes on the node to have their own communications paths.

It is important to note that some transport services do not implement all of the listed functionalities. For example, the UDP protocol does not offer connection control or reliable transfer.

The following are examples of protocols that provide message transport:

- SPX (Sequenced Packet eXchange)
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- NetBIOS/NetBEUI (Network Basic Input/Output System/NetBIOS Extended User Interface)
- APPC (Advanced Program-to-Program Communications)
- AppleTalk

Packet Forwarding/Internetworking Services

Packet Forwarding/Internetworking Services transfer data packets and manage the path that data takes through the network. They includes the following functionalities.

Fragmentation/Reassembly. The Packet Forwarding/Internetworking service divides an application message into multiple packets of a size suitable for network transmission. The individual packets include information to allow the receiving node to reassemble them into the message. The service also validates the integrity of received packets and buffers, reorders, and reassembles packets into a complete message.

Addressing. The Packet Forwarding/Internetworking service encapsulates packets with addressing information.

Routing. The Packet Forwarding/Internetworking service can maintain routing information (a view of the network topology) that is used to determine the best route for each packet. Routing decisions are made based on the cost, percent utilization, delay, reliability, and similar factors for each possible route through the network.

Switching. Switching is the process of receiving a packet, selecting an appropriate outgoing path, and sending the packet. Switching is performed by routers and switches within the communications fabric. Switching can be implemented in several ways.

- For some network protocols (e.g., TCP/IP), routers draw upon dynamic routing information to switch packets to the appropriate path. This capability is especially important when connecting independent networks or subnets.
- For other network protocols (e.g., Ethernet, Token Ring), switching simply directs packets according to a table of physical addresses. The switch can build the table by "listening" to network traffic and determining which network nodes are connected to which switch port. Some protocols such as Frame Relay involve defining permanent routes (permanent virtual circuits, or PVCs) within the network. Because Frame Relay is switched based upon PVCs, routing functionality is not required.

Multicasting. The Packet Forwarding/Internetworking service may support multicasting, which is the process of transferring a single message to multiple recipients at the same time. Multicasting allows a sender to transfer a single copy of the message to the communications fabric, which then distributes the message to multiple recipients.

The following are examples of protocols that provide Packet Forwarding/Internetworking:

- IP (Internet Protocol)
- IP Multicast (emerging standard that uses a predefined set of IP addresses to instruct network routers to deliver each packet to all users involved in a multicast session)
- IPX (Internetwork Packet Exchange)
- ATM (Asynchronous Transfer Mode)
- Frame Relay
- X.25

The following are examples of network components that perform Packet Forwarding/Internetworking:

- Routers
- Switches
- ATM switches, Frame Relay switches, IP switches, Ethernet switches, etc.

The following are examples of protocols that maintain routing information tables within routers:

Distance Vector Protocols. Each router periodically informs neighboring routers as to the contents of routing table (destination addresses and routing metrics); routing decisions are made based on the total distance and other "costs" for each path:

- IP and IPX Routing Information Protocols (RIP)
- AppleTalk Routing Table Management Protocol (RTMP)
- Cisco's Interior Gateway Routing Protocol (IGRP) and Enhanced IGRP

Link-State Protocols. Each router periodically broadcasts changes to the routers directly on adjacent networks:

- Open Shortest Path First (OSPF)
- ISO's Intermediate System to Intermediate System (IS-IS)
- Novell's NetWare Link Services Protocol (NLSP)

Policy Routing Protocols. These allow Internet backbone routers to accept routing information from neighboring backbone providers on the basis of contracts or other nontechnical criteria; routing algorithms are Distance Vector:

- Border Gateway Protocol (BGR)
- Interdomain Routing Protocol (IDR)

Circuit Switching

While Message Transport services and Packet Forwarding/Internetworking services support the transfer of packetized data, Circuit Switching services establish physical circuits for the transfer of circuit-switched multimedia- and image-oriented content such as voice, fax, video.

Circuit Switching Packetized uses an end-to-end physical connection between the sender and the receiver that lasts for the duration of the "call" transferred through brief, temporary, logical connections between nodes.

Circuit Switching services include the following functionality:

- Establishing end-to-end path for circuit (may involved multiple intermediate nodes/switches)
- Managing end-to-end path (quality, billing, termination, etc.)

The following are examples of Circuit Switching services:

- Analog dial-up telephone circuit
- Cellular telephone circuit
- ISDN (Integrated Services Digital Network)

Transport Security

Transport Security services (within the Transport Services layer) perform encryption and filtering.

Transport-Layer Encryption. Encryption within the Transport Services layer is performed by encrypting the packets generated by higher level services (e.g., Message Transport) and encapsulating them in lower level packets (e.g., Packet Forwarding/Internetworking). (Note that encryption can also occur within the Communications Services layer or the Network Media Services layer.) Encryption within the Transport Services layer has the advantage of being independent of both the application and the transmission media, but it may make network monitoring and troubleshooting activities more difficult.

The following standards support transport-layer encryption:

- Point to Point Tunneling Protocol
- Layer 2 Tunneling Protocol

Transport-layer Filtering. Network traffic can be controlled at the Transport Services layer by filtering data packets based on source and/or destination addresses and network service. This ensures that only authorized data transfers can occur. This filtering is one of the roles of a packet filtering firewall. (A firewall is a system that enforces an access control policy between a trusted internal network and an untrusted external network.)

The IETF standard IPSec supports interoperability among security systems. IPSec allows two nodes to dynamically agree on a security association based on keys, encryption, authentication algorithms, and other parameters for the connection before any communications take place; operates in the IP layer and supports TCP or UDP. IPSec will be included as part of IPng, or the next generation of IP (IPv6).

Network Address Allocation Services

Network Address Allocation services manage the distribution of addresses to network nodes. This provides more flexibility compared to having all nodes assigned static addresses. This service assigns addresses to nodes when they initially power-on and connect to the network.

The following are examples of standards that implement Network Address Allocation and allow a network node to ask a central resource for the node's network address (e.g., IP address):

- DHCP (Dynamic Host Configuration Protocol)
- BootP (Bootstrap Protocol)

Quality of Service Services

Different types of network traffic (e.g., data, voice, and video) have different quality of service requirements. For example, data associated with video conferencing sessions is useless if it is not delivered "on time." On the other hand, traditional best-effort data services, such as file or e-mail transfer, are not affected by variations in latency. Quality of Service (QoS) services deliver a defined network throughput for designated traffic by allocating dedicated bandwidth, prioritizing data traffic, etc. (Note that, as an alternative to predefined throughput, some QoS protocols can also offer a best effort, i.e., variable) throughput QoS based on available network capacity.)

Exhibit 24. Quality of Service Parameters

Parameter	Description
Connection establishment delay	Time between the connection request and a confirm being received by the requester
Connection establishment failure probability	Chance that the connection will not be established within the maximum establishment delay
Throughput	Bits per second of transmitted data
Transit delay	Time elapsed between when sender transfers packet and recipient receives packet
Residual error rate	Number of lost or corrupted messages compared to total messages in the sampling period.
Transfer failure probability	The fraction of the time when the throughput, transit delay, or residual error were not those agreed upon at the start of the connection.
Connection release delay	Time between when one node initiates a release and the other node performs the release
Connection release failure probability	Fraction of release attempts which do not succeed
Protection	Specifies a secure connection
Priority	Indicates traffic priority over the connection
Resilience	Probability that the transport layer spontaneously terminates

Exhibit 24 provides a description of various Quality of Service parameters.

Quality of Service can be achieved in various ways.

Specialized QoS Communications Protocols. These provide guaranteed QoS.

Asynchronous Transfer Mode (ATM). ATM is a connection-oriented wide area and local area networking protocol that delivers QoS on a per-connection basis. QoS is negotiated as part of the initial connection set up and as network conditions change. Because of the small size of ATM data cells, QoS can be better managed, compared to protocols such as Ethernet that have large frames that can tie up network components. For ATM to deliver QOS to applications, ATM must be used end to end.

Resource Reservation Protocol (RSVP). The emerging RSVP specification, proposed by the Internet Engineering Task Force (IETF), allows applications to reserve router bandwidth for delay-sensitive IP traffic. With RSVP, QoS is negotiated for each application connection. RSVP enables the network to reserve resources from end to end, using Frame Relay techniques on Frame Relay networks, ATM techniques on ATM, and so on. In this way, RSVP can achieve QoS across a variety of network technologies, as long as all intermediate nodes are RSVP capable.

IP Stream Switching. This improves network performance but does not guarantee QoS.

IP Switching. IP Switching is an emerging technology that can increase network throughput for streams of data by combining IP routing software with ATM switching hardware. With IP Switching, an IP switch analyzes each stream of packets directed from a single source to a specific destination and classifies it as short or long lived. Long-lived flows are assigned ATM Virtual Channels (VCs) that bypass the IP router and move through the switching fabric at the full ATM line speed. Short-lived flows continue to be routed through traditional store-and-forward transfer.

Tag Switching. Like IP Switching, emerging Tag Switching technology also improves network throughput for IP data streams. Tag Switching aggregates one or more data streams destined for the same location and assigns a single tag to all associated packets. This allows routers to more efficiently transfer the tagged data. Tag Switching is also known as Multiprotocol Label Switching.

Data Prioritization. This improves network performance for prioritized application traffic but does not guarantee QoS.

Although not an example of end-to-end QoS, various network components can be configured to prioritize their handling of specified types of traffic. For example, routers can be configured to handle legacy mainframe traffic (SNA) in front of other traffic (e.g., TCP/IP). A similar technique is the use of prioritized circuits within Frame Relay, in which the Frame Relay network vendor assigns different priorities to different permanent virtual circuits.

Prioritization techniques are of limited effectiveness if data must also pass through network components that are not configured for prioritization (e.g., network components run by third party network providers).

TRANSPORT SERVICES SUMMARY

Transport Services continue to improve and evolve to new levels. Through enhanced quality, tighter security, improved management and control, and increased speeds, transport services play an important role in moving key business information to an intended destination quickly, safely, and accurately. As netcentric computing continues to evolve, transport services should continue to converge to an infrastructure based on open industry standard technologies that integrate the many physical networking options available today. The next section discusses these physical networking options in more detail.

NETWORK MEDIA SERVICES

The Network Media layer, which provides the core of the communication fabric from the overall communications architecture framework, provides the following capabilities:

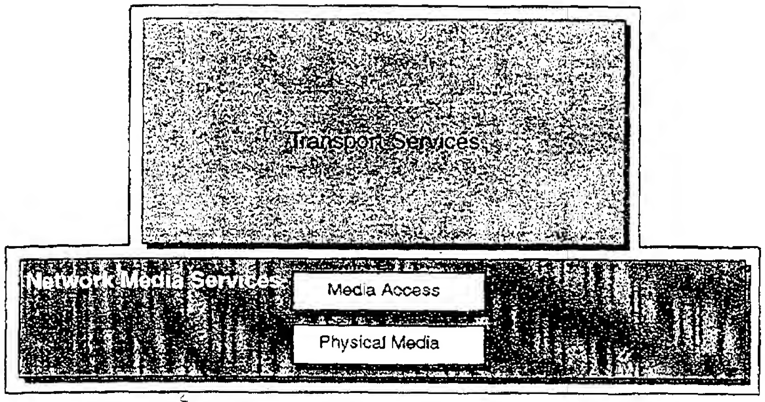

Exhibit 25. Network Media Services.

- Final framing of data for interfacing with the physical network
- Receiving, interpreting and acting on signals from the communications fabric
- Transferring data through the physical network

Network Media services (Exhibit 25) performs two primary service functions:

- Media Access services
- Physical Media services

Media Access Services

Media Access services manage the low-level transfer of data between network nodes. Media Access services perform the following functions.

Physical Addressing. The Media Access service encapsulates packets with physical address information used by the data link protocol (e.g., Ethernet and Frame Relay).

Packet Transfer. The Media Access service uses the data link communications protocol to frame packets and transfer them to another computer on the same network/subnetwork.

Shared Access. The Media Access service provides a method for multiple network nodes to share access to a physical network. Shared Access schemes include the following.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection). A method by which multiple nodes can access a shared physical media by "listening" until no other transmissions are detected and then transmitting and checking to see if simultaneous transmission occurred.

Token passing. A method of managing access to a shared physical media by circulating a token (a special control message) among nodes to designate which node has the right to transmit.

Multiplexing. A method of sharing physical media among nodes by consolidating multiple, independent channels into a single circuit. The independent channels (assigned to nodes, applications, or voice calls) can be combined in the following ways.

Time division multiplexing (TDM) — use of a circuit is divided into a series of time slots, and each independent channel is assigned its own periodic slot.

Frequency division multiplexing (FDM) — each independent channel is assigned its own frequency range, allowing all channels to be carried simultaneously.

Flow Control. The Media Access service manages the flow of data to account for differing data transfer rates between devices. For example, flow control would have to limit outbound traffic if a receiving machine or intermediate node operates at a slower data rate, possibly due to the use of different network technologies and topologies or due to excess network traffic at a node.

Error Recovery. The Media Access service performs error recovery, which is the capability to detect and possibly resolve data corruption that occurs during transmission. Error recovery involves the use of checksums, parity bits, etc.

Encryption. The Media Access service may perform encryption. (Note that encryption can also occur within the Communications Services layer or the Transport Services layer.) Within the Network Media Services layer, encryption occurs as part of the data link protocol (e.g., Ethernet, frame relay). In this case, all data are encrypted before it is placed on the wire. Such encryption tools are generally hardware products. Encryption at this level has the advantage of being transparent to higher level services. However, because it is dependent on the data link protocol, it has the disadvantage of requiring a different solution for each data link protocol.

The following are examples of Media Access protocols:

- Ethernet
- Token Ring

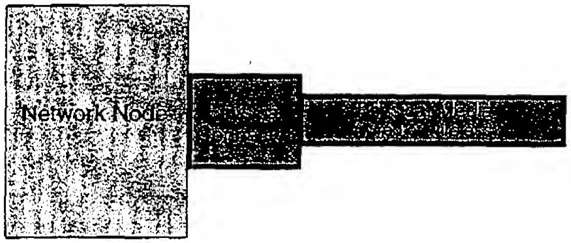

Exhibit 26. Subcomponents of Physical Media.

- FDDI (Fiber Distributed Data interface)
- Portions of the ATM (Asynchronous Transfer Mode) standard
- HDLC (High-level Data Link Control)/SDLC (Synchronous Data Link Control)
- LAP-B (Link Access Procedure — Balanced)
- T-carrier, E-carrier (e.g., T1, T3, E1, E3)
- TDM and FDM (Time Division Multiplexing and Frequency Division Multiplexing; used on T-carriers, etc.)
- SONET (Synchronous Optical Network), SDH
- PPP (Point-to-Point Protocol), SLIP (Serial Line Internet Protocol)
- V.32, V.34, V.34 bis, etc.
- RS-232, EIA-232
- TDMA and FDMA (Time Division Multiple Access and Frequency Division Multiple Access; used on wireless links)

Specialized services convert between addresses at the Media Access level (i.e., physical addresses like Ethernet) and the Packet Forwarding/Internetworking level (i.e., network addresses like IP). The following protocols are examples of this functionality.

Address Resolution Protocol (ARP). ARP allows a node to obtain the physical address for another node when only the IP address is known.

Reverse Address Resolution Protocol (RARP). RARP allows a node to obtain the IP address for another node when only the physical address is known.

Physical Media Services

The Physical Media are divided into two categories (Exhibit 26):

- Physical connectors
- Physical media (wired or wireless)

Physical Connectors. The following are examples of wiring connectors used to connect network nodes to physical media:

- RJ-11, RJ-45
- BNC
- DB-9, DB-25
- Fiber optic connectors

Physical Media. Physical Media may be wired or wireless. Wired Physical Media includes wiring and cabling, while wireless Physical Media includes antennas, connectors, and the radio frequency spectrum.

The following are examples of wired physical media:

- Twisted pair wiring
- Shielded twisted pair wiring
- Coaxial cable
- Fiber optic cable
- Four-pair voice-grade wiring

The following are examples of wireless physical media:

- Cellular antennas and the associated radio frequencies
- Wireless local area network antennas and the associated radio frequencies
- Satellite antennas and the associated radio frequencies

NETWORK MEDIA SERVICES SUMMARY

Without the Network Media Services (which we compared earlier to the interconnected train tracks, signals, and switches), information would not be capable of traveling to its intended destinations. While this infrastructure is a complex network of numerous interconnected copper wires, fiber optics cables, and radio antennas, continued change in Network Media Services is likely to be slow. We are more likely to continue to see new technologies evolve to adapt and bridge the various physical network options. These technologies make up the essense of netcentric computing, which continues to expand the reach of client/server while delivering rich new content.

CONCLUSION

Today's advanced communications architectures permit organizations to take full advantage of the convergence of computing, communications, and knowledge. Netcentric computing applications provide more direct links with business partners and allow companies to respond quickly to fluctuations in customer demand. As communications architectures grow in sophistication, one should expect the network to enable almost total supply chain integration. Applications that manage and perform business-to-business processes will enable the ultimate virtualization of business: bringing together strategy, people, process, and technology in a unique configuration across multiple companies to serve the customer in a more powerful way than any one company could on its own. That will be the final convergence, one in which most barriers between companies and their customers have been removed.